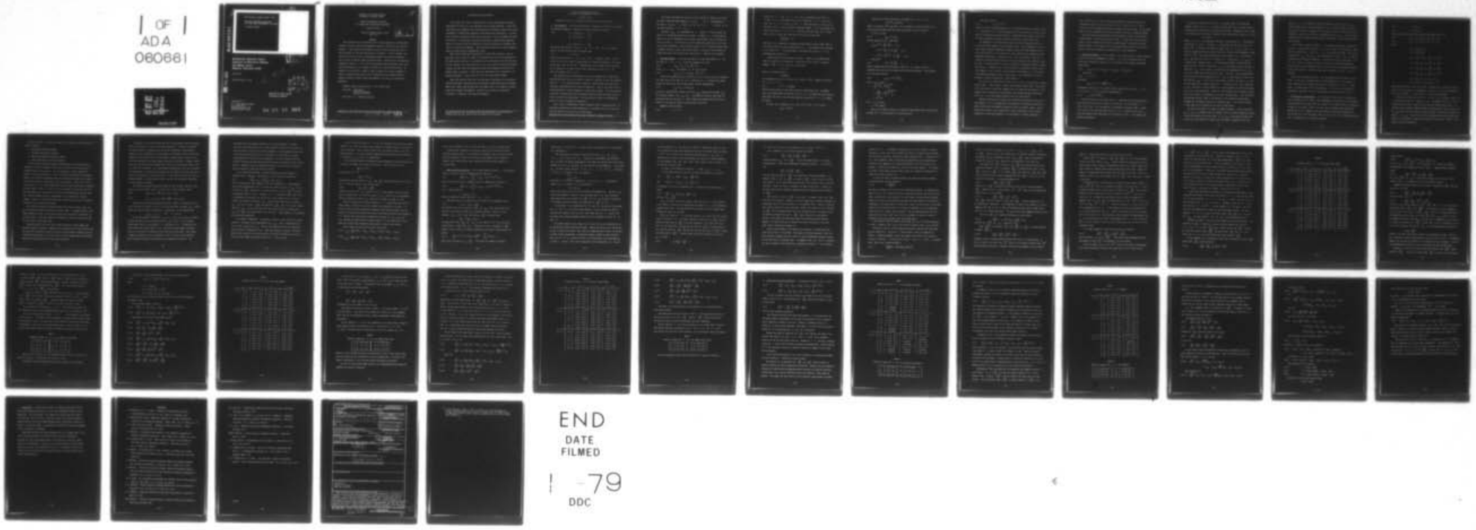
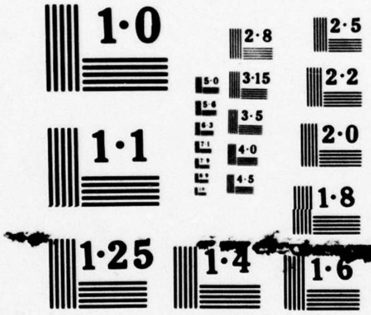
AD-A060 661 WISCONSIN UNIV-MADISON MATHEMATICS RESEARCH CENTER F/G 12/1
SOLVING DIFFERENTIAL EQUATIONS ON A HAND HELD PROGRAMMABLE CALC--ETC(U)
APR 78 J B ROSSER DAAG29-75-C-0024
UNCLASSIFIED MRC-TSR-1848 NL





NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

AD A060661

MRC Technical Summary Report #1848

SOLVING DIFFERENTIAL EQUATIONS ON A HAND HELD PROGRAMMABLE CALCULATOR

J. Barkley Rosser

LEVEL

Mathematics Research Center
University of Wisconsin–Madison
610 Walnut Street
Madison, Wisconsin 53706

DDC FILE COPY

April 1978

Received April 6, 1978

See 1473

DDC
RECEIVED
NOV 1 1978
A

Approved for public release
Distribution unlimited

Sponsored by

U. S. Army Research Office
P.O. Box 12211
Research Triangle Park
North Carolina 27709

78 09 12 051

UNIVERSITY OF WISCONSIN-MADISON
MATHEMATICS RESEARCH CENTER

ACCESSION for
NTIS White Section ☑
DDC Buff Section □
UNANNOUNCED □
JUSTIFICATION
BY
DISTRIBUTION/AVAILABILITY CODES
Dist. AVAIL. and/or SPECIAL
A

SOLVING DIFFERENTIAL EQUATIONS ON A HAND HELD PROGRAMMABLE CALCULATOR

J. Barkley Rosser

Technical Summary Report #1848
April 1978

ABSTRACT

Most scientists who occasionally have to solve numerically a differential equation now own a hand held programmable calculator which will very often be adequate. Since hand held calculators are slow, there is particular need to keep the number of function evaluations to a minimum. At first thought, this would seem to rule out use of Runge-Kutta methods, but recent developments, such as those by Fehlberg (mostly unknown except to specialists), may make them competitive after all. In the area of predictor-corrector methods, some variations make excessive use of memory locations for a hand held calculator. An analysis of such matters is made in order to advise as to good procedures to follow, including alerting the solver to methods that are seldom taught in numerical analysis courses (where the emphasis is on the use of large fast computers).

AMS(MOS) Subject Classification: 34-02, 34A50, 65L05

Key Words: Runge-Kutta
predictor-corrector
numerical stability

Work Unit No. 7 - Numerical Analysis

Sponsored by the United States Army under Contract No. DAAG29-75-C-0024.

78 09 12 051

Significance and Explanation

Until about 1950, when a scientist had to solve a differential equation numerically he had to do it by laborious use of a desk calculator. Since that time large scale computers have revolutionized the situation because they can be programmed to perform fast repetitive calculations very efficiently. In recent years hand-held calculators have come on the scene. They are adequate for many calculations, in particular for solving some kinds of differential equations. However the factors that are important for hand calculators are different from those for large scale computers. Hand calculators have only a limited number of memory locations and are comparatively slow.

When solving differential equations on a hand held calculator, there is particular need to keep the number of calculations of the function appearing in the differential equation to a minimum, else the calculation will take too long. At first thought, this would seem to rule out use of what are called Runge-Kutta methods, but some recent developments may make them competitive after all for certain types of problems. These types are identified, and a discussion, with numerical examples, is given how best to use the Runge-Kutta methods.

Some other methods which call for much fewer calculations of the function require more memory locations than are available on many hand held calculators. There still remain some methods which are modest both in the number of calculations and in memory requirements. How best to use these on a hand held calculator is explained, with numerical examples.

The responsibility for the wording and views expressed in this descriptive summary lies with MRC, and not with the author of this report.

SOLVING DIFFERENTIAL EQUATIONS ON A HAND HELD PROGRAMMABLE CALCULATOR

J. Barkley Rosser

Dedicated to Prof. Dr. Johannes Weissinger on his 65th birthday.

1. Preliminaries. The present discussion is limited to initial value ordinary differential equations. One wishes to solve a system of equations

$$\begin{aligned} y_1' &= f_1(x,y_1,y_2, \ldots, y_p) \\ y_2' &= f_2(x,y_1,y_2, \ldots, y_p) \\ &\ldots\ldots\ldots\ldots\ldots \\ y_p' &= f_p(x,y_1,y_2, \ldots, y_p), \end{aligned} \tag{1.1}$$

being given the values of $y_1, y_2, \ldots, y_p$ at $x = x_0$; here y_i' denotes dy_i/dx. We will discuss only the special case

$$y' = f(x,y) , \tag{1.2}$$

being given the value of y at $x = x_0$; here y' denotes dy/dx. The discussion of (1.2) can easily be generalized to a system of equations. See Conte and de Boor, 1972, pp. 365-366. Incidentally, sets of higher order equations can be reduced to a system of first order equations, such as (1.1). See Conte and de Boor, 1972, p. 365.

The person who only occasionally has to solve numerically a differential equation may never have had a course in numerical analysis, or may have forgotten much of it. So it seems necessary to make the present paper reasonably self-contained, presupposing little previous experience. Even if the reader has previous experience, it was presumably on a large fast computer. For the hand held calculator, considerations are sufficiently different that one cannot rely too much on such previous experience. The discussion to follow is comprehensive enough to cover the important differences.

Also, there is disagreement between reputable texts on various points. An adjudication between them, with explanations, is included, in spite of the fact that this adds to the bulk of the present paper.

Sponsored by the United States Army under Contract No. DAAG29-75-C-0024.

The overall procedure for solution is the classic one, namely to try step by step to calculate approximations $y_1, y_2, y_3, \ldots$ for y corresponding to the values $x_1, x_2, x_3, \ldots$, where $x_0 < x_1 < x_2 < x_3 < \ldots$. We take y_0 as the value given for y at $x = x_0$.

We denote $x_{n+1} - x_n$ generically by h . Usually h will be taken the same for a considerable succession of steps. A change in h is occasionally called for, but not uncommonly this involves some travail, and in the main one tries to avoid it. On the other hand, all authorities agree that at each step (or at least frequently), tests should be made to see if a change of step length is needed, shorter to keep errors under control, or longer to avoid unduly extended calculation that would result from taking h smaller than necessary.

2. The Euler method. If one has proceeded to a good approximation y_n for the value of y at $x = x_n$, a Taylor's expansion will give

$$y_{n+1} = y_n + h\, y'_n + \frac{h^2}{2} y''(\xi), \tag{2.1}$$

where $x_n < \xi < x_{n+1}$. We evaluate y'_n by (1.2), and get

$$y_{n+1} = y_n + h\, f(x_n, y_n) + \frac{h^2}{2}\; y''(\xi) \; . \tag{2.2}$$

Unless one already knows the solution of (1.2), one has no way to calculate the final term on the right of (2.2). Certainly, if h is taken to be small enough, the final term will be quite small, and the approximation

$$y_{n+1} \cong y_n + h\, f(x_n, y_n) \tag{2.3}$$

will give a sufficiently accurate value for y_{n+1}, after which one gets y_{n+2} by a similar formula, then y_{n+3}, etc. A problem with which we must cope is being sure that we have taken h small enough that repeated use of (2.3) instead of (2.2) does not lead to serious error in the end.

Suppose we wish to solve

$$y' = -y, \tag{2.4}$$

given that $y = 1$ when $x = 0$. Let us try to approximate the value of y when $x = 6$. As the answer for (2.4) is obviously $y = e^{-x}$, we can say that for $0 < \xi \leq 6$ we have $|y''(\xi)| < 1$. Hence the error in (2.3) will be less than $h^2/2$. If we take steps of constant length h, we will require $6/h$ steps to get from $x = 0$ to $x = 6$. With an error less than $h^2/2$ at each step, and $6/h$ steps, the final error will add up to less than

$$\left(\frac{h^2}{2}\right)\left(\frac{6}{h}\right) = 3h .$$

From this, it is tempting to conclude that the method is first order; that is, the overall error is roughly proportional to h. For example, if we decrease h by a factor of 2, we would expect to decrease the error at $x = 6$ by approximately a factor of 2.

Within bounds, the conclusion is correct. However, the argument given above to support it is fallacious. To see this, let us look at a specific example. Take $h = 0.1$. By (2.3), we will get

$$y_1 = 0.9.$$

This is too small by about

$$0.0048374.$$

(In accordance with (2.2), this error is close to $h^2/2$.) However, the value of y at $x = 6$ is

$$e^{-6} \cong 0.0024888. \tag{2.5}$$

Thus our final answer is less than the error on the first step. To suggest that we can approximate the overall error at $x = 6$ by adding up such errors as shown above for each of the 60 steps required to get from 0 to 6 is simply not sound.

In fact, for the equation (2.4), use of (2.3) with $h = 0.1$ gives

$$y_{n+1} = (0.9)y_n .$$

Using this 60 times would give an estimate for y at $x = 6$ of

$$(0.9)^{60} \cong 0.0017970.$$

This is in error by less than 28%. For a procedure in which the error on the first step was almost twice the total final answer, this is not bad.

By (2.3), we get

$$y_{n+1} \cong (1-h)y_n \tag{2.6}$$

for the equation (2.4). This gives

$$\begin{aligned} y_{n+1} &\cong \{e^{-h} - \frac{h^2}{2!} + \frac{h^3}{3!} - \dots\} y_n \\ &= \{e^{-h} - \frac{h^2}{2}(1 - \frac{h}{3} + \frac{h^2}{12} - \dots)\}y_n \\ &= e^{-h}\{1 - \frac{h^2}{2} e^h (1 - \frac{h}{3} + \frac{h^2}{12} - \dots)\}y_n \,. \end{aligned}$$

The factor

$$e^h(1 - \frac{h}{3} + \frac{h^2}{12} - \dots) \tag{2.7}$$

tends to remain constant and close to unity for $|h|$ small, since as h increases the first factor increases and the second decreases. So we replace (2.7) by unity, getting

$$y_{n+1} \cong e^{-h}\{1- \frac{h^2}{2}\} y_n \,. \tag{2.8}$$

So at $x = 6$, we get

$$\begin{aligned} y &= \left[e^{-h}\left\{1 - \frac{h^2}{2}\right\}\right]^{6/h} \\ &= e^{-6}\left[\left\{1 - \frac{h^2}{2}\right\}^{-\frac{2}{h^2}}\right]^{-3h} \\ &\cong e^{-6}\, e^{-3h} \,. \end{aligned}$$

For $h = 0.1$, we get

$$y = e^{-6}\{0.74082\}.$$

Hence, by the above analysis, we expect the calculated value to be too low by about 26%: it was actually too low by about 28%.

The final formula,

$$y \cong e^{-6} e^{-3h} \tag{2.9}$$

shows that (for a reasonable range of h) the relative error is indeed of order h. Actually for $h = 0.1$ and the equation (2.4), we see by (2.2) that we have about 0.5% relative error at each step. Accumulating such relative errors for 60 steps could give an overall error of 30%, which is about what we got.

So sometimes it is absolute error that one should accumulate from step to step, but other times it can be relative error. For programmers who try to write general purpose differential equation solvers for large computers with error control built in, the question of how to handle error accumulation poses formidable difficulties. See Hull, et al., 1972, pp. 607-608. When one is solving on a hand held calculator, one sees the progress of the solution, step by step. In regions where it is best to reckon by accumulating relative errors, one can do so. However, when it would be better to accumulate absolute errors (for example, if one is going to pass through a zero value of y), the change to accumulating absolute errors is easily made. Such flexibility is hard to arrange in a preset program for a large computer.

By (2.9), if we should wish to get to $x = 6$ with 0.1% accuracy, we should have to take h about 1/3000. Thus we would require 18,000 steps. This would not be too bad on a large fast computer, but on a hand held calculator it could require hours, especially if our equation to be solved were more complicated than (2.4). So we need something better than the Euler method.

Actually, there were two reasons for including the present section on the Euler method. One is to demonstrate the importance of maintaining flexibility about whether one is accumulating relative errors or absolute errors. With a hand held calculator, such flexibility is easily maintained. With a preset program for a large fast computer, such flexibility is almost impossible.

Thus, in Hull, et al., 1972, the decision was made to lock one's self into accumulating absolute errors (see their p. 607). To get to $x = 6$ with a 0.1% error by this scheme would require far more than 18,000 steps. On the other hand, the large computers are so very fast that it is still a practical scheme.

The other reason for mentioning the Euler method is that it is useful for "looking ahead." Using a few steps with large h takes very little time and gives at least a general idea of what one might expect to encounter for some distance ahead. If one takes such a "look ahead" every so often it can help in planning how best to arrange the upcoming part of the integration.

3. Runge-Kutta methods. The text Henrici, 1977, explains how to do many calculations on the HP-25 programmable calculator. On p. 182, he suggests that for solving the differential equation (1.2) one might use the second order Runge-Kutta method

$$(3.1) \qquad y_{n+1} \cong y_n + \frac{h}{2}\{f(x_n,y_n) + f(x_{n+1},y_n + h\,f(x_n,y_n))\}.$$

Applied to

$$(3.2) \qquad y' = k\,y\,,$$

this gives

$$y_{n+1} \cong \left(1 + hk + \frac{(hk)^2}{2}\right) y_n\,.$$

An analysis like that in the previous section shows that if we set $y = 1$ at $x = 0$, then at $x = X$ we would get approximately

$$(3.3) \qquad y \cong e^{kX}\, e^{-h^2k^3X/6} \quad .$$

The first factor on the right is what y should be, and the second factor shows about how far off we are from the true value. So, if we take $k = -1$ (as in (2.4)) and $X = 6$, and ask for 0.1% accuracy, we need to take h about 1/32. Thus it will take 192 steps to get to $x = 6$. However (note (3.1)), each step requires TWO evaluations of the function $f(x,y)$. So we require 384 function evaluations.

Actually, we require more than that. As stated above, one should make frequent checks to see if h is about the right size; this is a point that we failed to come to grips with in the previous section. Analogous to (2.2), there is a formula (see pp. 192-200 of Ralston, 1965) that gives the error of (3.1) as

$$h^3K + h^4L + \dots ,$$

where K, L, etc., are complicated functions of derivatives of y and partial derivatives of $f(x,y)$. For small h, we may say that the error of (3.1) is about h^3K. If we take two successive steps, from y_n to y_{n+2}, and if K does not change much from one step to the next, then we could accumulate a local error of $2h^3K$ (over and above whatever error we had at y_n) in getting to y_{n+2}. Now apply (3.1) with $2h$ in place of h to get from y_n to y_{n+2} in a single step. Assuming that K is not flucuating badly, we would make an error of about $(2h)^3K$ in getting to y_{n+2}, or four times the error we made in getting to y_{n+2} in two steps. So now we have two approximations for y_{n+2}, one with an error about four times the other. From this, we can estimate about how far off our better approximation is. If it is close enough, we take it as the value of y_{n+2}. If it is not close enough, we have to go back to y_n and try over again with a smaller value of h. See Hull, et al., 1972, bottom of p. 616. How close is "close enough" is a sticky question for which there seems not to be a very good answer. After all, this involves only two steps, and one has to worry about the accumulation of errors over many steps. And, as noted in the previous section, there is the question if one should be worried about accumulation of absolute errors or of relative errors. But at least one has to have some sort of estimate of the step by step errors.

As pointed out above, if we take two steps of length h each to get from y_n to y_{n+2}, we probably have an estimate that is in error by about one fourth of what we would get if we went from y_n to y_{n+2} in one step of length $2h$. If the error were EXACTLY one fourth, we could write a formula for the true value of y. It is tempting to take this formula as the value for y_{n+2}. It most likely gives a better estimate for y_{n+2}. However, there is no way to say how much better; an estimate for the step by step error would then not be available. Also, one is left with no improved value for y_{n+1}. So we had best consider that this formula merely provides an estimate of the error after each pair of steps.

For the pair of steps of length h, we required two function evaluations per step. For the step of length $2h$, we also require two evaluations, except that the evaluation of $f(x,y)$ at $x = x_n$ has already been made. So, for the evaluation of y_{n+1} and y_{n+2}, with an estimate of error at x_{n+2}, we require altogether five function evaluations. So for the 192 steps to get from $x = 0$ to $x = 6$, that is, 96 pairs of steps, we require 480 function evaluations. While this is a great improvement over the 18,000 function evaluations required by the Euler method, it could be rather time consuming if $f(x,y)$ is at all complex.

So we wish for something better. We cannot manage anything better on the HP-25. It has only 50 program steps, and implementation of (3.1) uses 39 of these, leaving only 11 program steps for the calculation of $f(x,y)$. In many cases, 11 program steps will not be adequate. So, if we are to do much with differential equations, we need a more capable calculator than the HP-25.

4. Alternatives to Runge-Kutta. Let us suppose that we have a calculator at least as capable as the HP-65. Most programmable calculators now on the market are appreciably more capable than the HP-65, but it sufficed for the calculations recorded in this paper. With it, one can carry out higher order Runge-Kutta methods than (3.1). Suppose we use the classical fourth order Runge-Kutta. See (5.6-48) on p. 200 of Ralston, 1965, or (6.37) on p. 338 of Conte and de Boor, 1972. Incidentally, if we have two equations

(4.1) $$y' = f(x,y,z)$$

(4.2) $$z' = g(x,y,z),$$

then the classical fourth order Runge-Kutta method consists of

(4.3) $$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

(4.4) $$z_{n+1} = z_n + \frac{1}{6}(\ell_1 + 2\ell_2 + 2\ell_3 + \ell_4)$$

where

$$k_1 = h\, f(x_n, y_n, z_n)$$

$$\ell_1 = h\, g(x_n, y_n, z_n)$$

$$k_2 = h\, f(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}, z_n + \frac{\ell_1}{2})$$

$$\ell_2 = h\, g(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}, z_n + \frac{\ell_1}{2})$$

$$k_3 = h\, f(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}, z_n + \frac{\ell_2}{2})$$

$$\ell_3 = h\, g(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}, z_n + \frac{\ell_2}{2})$$

$$k_4 = h\, f(x_n + h, y_n + k_3, z_n + \ell_3)$$

$$\ell_4 = h\, g(x_n + h, y_n + k_3, z_n + \ell_3).$$

See Conte and de Boor, 1972, pp. 365-366. Extension to a system of equations such as (1.1) is obvious. Incidentally, the error in (4.3) is approximately h^5K for a suitable K, and the error in (4.4) correspondingly.

If we wish to solve (2.4) out to $x = 6$ with 0.1% accuracy, we must take h slightly less than 3/8. Taking $h = 3/8$ is probably close enough, which would require 16 steps. There are four function evaluations per step. To accomplish the procedure suggested above for estimating the step by step error of y_{n+2} after a pair of steps, we must also make a single step of length $2h$ from y_n to y_{n+2}. This also takes four evaluations, but one has been done before. So we require 11 function evaluations for each pair of steps. So we need 88 function evaluations.

Can we do better? In Enright and Hull, 1976, testing is reported of four sorts of methods:

1. Runge-Kutta-Fehlberg methods.
2. Rational extrapolation methods.
3. Adams type predictor-corrector methods.
4. Milne type predictor-corrector methods.

The Runge-Kutta-Fehlberg methods are Runge-Kutta type methods improved according to ideas in Fehlberg, 1968, 1969, and 1970,to give easier ways of estimating step by step errors. This is purported to reduce somewhat the number of function evaluations. In view of certain advantages of Runge-Kutta methods, the Fehlberg improvements make the Runge-Kutta methods fairly competitive when the function evaluations can be done quickly. However on p. 949 of Enright and Hull, 1976, some disadvantages with the Fehlberg methods are reported. By phone (March, 1978) T. E. Hull of Toronto informed the present writer that new improvements had eliminated certain of these disadvantages, and that IMSL has recently embodied these new improvements in its Runge-Kutta software package. However, at best these methods are competitive only when the function evaluations can be done quite quickly.

The rational extrapolation methods derive from ideas of Gragg, 1965, and were developed carefully in Bulirsch and Stoer, 1966. In Enright and Hull, 1976, these methods are given a fairly good rating when the function evaluations can be done quickly. They seem complex to program for a hand held calculator, and so we will say no more about them.

Enright and Hull, 1976, give their highest ratings to certain Adams type predictor-corrector methods. However, their conclusions do not necessarily hold for hand held calculators because of the very limited number of memory locations of hand held calculators. So we will make a study of Adams type and Milne type predictor-corrector methods for hand held calculators.

Incidentally, one cannot merely take one of the programs tested in Enright and Hull, 1976, and put it on his hand held calculator. In order to accomodate the wide diversity of vagaries that can arise in solutions of differential equations, these programs have a large amount of "overhead", and are beyond the capabilities of present hand held calculators, besides being so long as to be very time consuming at the slow speed of hand held calculators. However, unless one has a very large system of equations to solve, one can proceed step by step on a hand held calculator. One "looks ahead" periodically as an aid to planning the calculation, one monitors the accumulation of errors as one goes, and one is alert for idiosyncracies that might arise. If the latter do arise, one can try shifting methods; if worse comes to worst, one can try a change of variables, or more complex strategems.

The classical predictor-corrector method is that of Milne (see (28.1) and (28.2) on p. 65 of Milne, 1953, or (5.5-12) on p. 182 of Ralston, 1965):

$$y^{p}_{n+1} = y_{n-3} + \frac{4h}{3}(2y'_n - y'_{n-1} + 2y'_{n-2}) \tag{4.5}$$

$$y_{n+1} = y_{n-1} + \frac{h}{3}(f(x_{n+1}, y^{(p)}_{n+1}) + 4y'_n + y'_{n-1}). \tag{4.6}$$

This uses two function evaluations per step. There is the obvious one in (4.6), and when one comes to use (4.5) for the next step, one needs y'_{n+1}, which calls for a second function evaluation. Incidentally, it is required that

$$y_{n+1-i} - y_{n-i} = h \quad \text{for} \quad i = 0,1,2,3.$$

To use this method, we must have y_{n-3}, y_{n-1}, y'_{n-2}, y'_{n-1}, and y'_n stored, and for the next step we will have to have y_{n-2} and y_n. So our storage requirements amount to four values of y and three of y'. In addition, one has to carry the current value of x_n. This uses up eight memory locations, which for all practical purposes are all there are on the HP-65. This leaves no memory locations to use in evaluating $f(x,y)$. So use of the classical Milne predictor-corrector would often not be possible on the HP-65. For calculators with more memory locations, the large requirement for memory locations could preclude use of the Milne predictor-corrector if one wishes to solve a system of even as few as three equations. A more compelling reason not to use the Milne method is that it is now known to be unstable in certain circumstances which arise not too infrequently. We shall produce predictor-corrector methods that have much more modest memory requirements and are stable besides. Meanwhile the Milne predictor-corrector will be used to illustrate typical features of predictor-corrector methods.

As the name would indicate, a predictor-corrector method embodies a predictor and a corrector. The predictor is (4.5) and the corrector is

$$y_{n+1}^{(c)} = y_{n-1} + \frac{h}{3} (y'_{n+1} + 4y'_n + y_{n-1}), \tag{4.7}$$

which is nothing more than Simpson's rule for integrating y' approximately. The obvious disadvantage of (4.7) is that one needs the value of y'_{n+1} to get the value of $y_{n+1}^{(c)}$, whereas by (1.2) one needs the value of $y_{n+1}^{(c)}$ to get the corresponding value of y'_{n+1}. Actually, if h is reasonably small and $f(x,y)$ is reasonably well behaved, one can get out of this impasse by an iteration scheme; after making a guess for y'_{n+1}, successively substitute the current y'_{n+1} into (4.7) to get a $y_{n+1}^{(c)}$ and substitute $y_{n+1}^{(c)}$ into (1.2) to get a better value for y'_{n+1}. In the usual case that will arise, this will converge to a y'_{n+1} and $y_{n+1}^{(c)}$ that satisfy both (4.7) and (1.2). Unfortunately, it is prodigal with function evaluations.

So we adopt a compromise. A predictor is given, in this case (4.5), which produces a reasonably close approximation for y_{n+1}. This is substituted into (1.2) to get a guess for y'_{n+1}. This is then substituted into the corrector; the net result is embodied in (4.6). There the process is stopped. We do not have as much accuracy as we could get with a few more iterations, but we have held the total number of function evaluations to two per step.

Since we need four values of y to use the predictor (and at equally spaced values of x), these four values must somehow be obtained before we can start to use this process. On pp. 61-64 of Milne, 1953, is given a scheme to get four starting values. It is now generally agreed that the best way to get started is to calculate y_1, y_2, and y_3 by Runge-Kutta.

One attractive feature of predictor-corrector methods is the ease with which one gets an estimate of the step by step error. The error for (4.5) is

$$\frac{14}{45} h^5 y^{(v)}(\xi)$$

and that for (4.7) is

$$-\frac{1}{90} h^5 y^{(v)}(\xi),$$

provided that one has values of y'_{n+1} and $y^{(c)}_{n+1}$ that satisfy both of (4.7) and (1.2). Then the error in (4.7) will be about

$$-\frac{1}{29} \{y^{(c)}_{n+1} - y^{(p)}_{n+1}\} . \tag{4.8}$$

Actually, the value of y_{n+1} given by (4.6) is close enough to what one would get by iterating with (4.7) that the error in y_{n+1} from (4.6) is approximately what one would get by using y_{n+1} from (4.6) in place of $y^{(c)}_{n+1}$ in (4.8).

Since one needs four consecutive values of y at equal step sizes for the Milne method, a change of step size (should it be required) is not easy. If one has as many as seven preceding consecutive steps of equal length, one can double the step size by picking every other value from the present and preceding y's. However this would require recovering the values of y_{n-6}, y_{n-4}, and y'_{n-4} besides the values that one usually stores. Or one could carry on for three more steps before doubling, being careful to save the key values.

If one wishes to halve the step size, one can use

$$y_{n-\frac{1}{2}} = \frac{1}{128}\{45y_n + 72y_{n-1} + 11y_{n-2} + h(-9y'_n + 36y'_{n-1} + 3y'_{n-2})\} \tag{4.9}$$

$$y_{n-\frac{3}{2}} = \frac{1}{128}\{11y_n + 72y_{n-1} + 45y_{n-2} - h(3y'_n + 36y'_{n-1}, - 9y'_{n-2})\} \tag{4.10}$$

(see p. 208 of Hamming, 1962, or (A57) and (A58) on p. 451 of Rosser, 1967).

Unfortunately, doubling or halving the step size is often not the most efficient change to make. For a change of a different size, one can use interpolation formulas, of which (4.9) and (4.10) are samples, but it might be simpler just to make a fresh start, generating the next three values of y by Runge-Kutta.

5. Adams predictor-correctors. For the Adams method of order r, the predictor (which is commonly called an Adams-Bashforth formula) is

$$y^{(p)}_{n+1} = y_n + h \sum_{i=0}^{r-1} \alpha_i y'_{n-i} + K_p h^{r+1} y^{(r+1)}(\xi), \tag{5.1}$$

and the corrector (which is commonly called an Adams-Moulton formula) is

$$y^{(c)}_{n+1} = y_n + h \sum_{i=0}^{r-1} \beta_i y'_{n+1-i} - K_c h^{r+1} y^{(r+1)}(\xi). \tag{5.2}$$

The h that appears is

$$h = x_{n+1-i} - x_{n-i},$$

which is required to be the same for $i = 0, \ldots, r-1$.

The error term on the right of (5.2) is based on the assumption that

$$y'_{n+1} = f(x_{n+1}, y^{(c)}_{n+1}). \tag{5.3}$$

As indicated with (4.7), values of $y^{(c)}_{n+1}$ and y'_{n+1} that satisfy both (5.2) and (5.3) can usually be found by an iterative process. That is, if one chooses a $\bar{y}_{n+1}$ that is near the limiting $y^{(c)}_{n+1}$, and then forms

$$\bar{y}'_{n+1} = f(x_{n+1}, \bar{y}_{n+1}),$$

and substitutes this $\bar{y}'_{n+1}$ for y'_{n+1} on the right of (5.2), one will get a value nearer to $y^{(c)}_{n+1}$ than $\bar{y}_{n+1}$. Repetition of this converges to $y^{(c)}_{n+1}$. However, this is costly in function evaluations, and what is mostly done is to define

$$y^{(t)}_{n+1} = y_n + h \beta_0 f(x_{n+1}, y^{(p)}_{n+1}) + h \sum_{i=1}^{r-1} \beta_i y'_{n+1-i}, \tag{5.4}$$

after which one takes $y_{n+1} = y^{(t)}_{n+1}$. This holds the number of function evaluations to two per step. (One may think of the superscript "t" as standing for "traditional.")

The requirements for memory locations are one for y_n, r for y'_{n-i} (i = 0,...,r-1), and one for x_n. Actually, on a calculator like the HP-65, one can reduce the memory requirements to one fewer by judicious use of the stack. Suppose we are at x_n, and have y_n, but have not yet calculated y'_n. Let us have y'_{n-i} stored at memory location i, for i = 1, ..., r-1, with y_n at location r, and x_n at location r+1. We calculate

$$\sum_{i=1}^{r-1} \alpha_i y'_{n-i} ,$$

and store it in location 1, having for i = r-2, r-3, ..., 1 successively stored y'_{n-i} in location i+1. Now calculate

$$y'_n = f(x_n, y_n) .$$

Push an extra copy of this up in the stack to hold momentarily. Now add $\alpha_0 y'_n$ to what was stored in location 1, put the extra copy of y'_n into location 1, and proceed with the calculation of $y^{(p)}_{n+1}$ by (5.1). We now have in storage or in the stack everything needed for calculation of $y^{(t)}_{n+1}$ by (5.4).

In the above, it is assumed that the α's and β's are part of the program. The α's and β's are fairly simple numbers, so that this does not seem to overload the program, unless one has a very large value of r, in which case one hopes that additional memory locations will be available to store the α's and β's.

The Adams formulas of any order, with an error term, can be derived by use of the Newton backward difference formula. Details are given on pp. 340-342 and 350-351 of Conte and de Boor, 1972, where the Adams method of order 4 is derived. A rather diffuse explanation is scattered through a number of pages of Milne, 1953, but on p. 50 is a table from which the coefficients can be calculated up to order 9. Henrici, 1962, gives coefficients of the predictors on p. 194 and of the correctors on p. 199, both up to order 6. Neither the tables of Milne nor of Henrici give the error terms, but these can be derived easily by taking $y = x^{r+1}$ in (5.1) and (5.2), which will give the values of K_p and K_c. Correctors up to order 9, with error terms, can be got by a trivial modification of formulas (A2), (A4), (A7), (A12), (A20), (A26), (A34), and (A42) on pp. 446-449 of Rosser, 1967.

In order to get started, we need the r values $y_0, y_1, \ldots, y_{r-1}$.

For the Adams method of order two, we have the predictor and corrector

$$(5.5) \qquad y_{n+1}^{(p)} = y_n + \frac{h}{2}(3y'_n - y'_{n-1}) + \frac{5h^3}{12} y'''(\xi)$$

$$(5.6) \qquad y_{n+1}^{(c)} = y_n + \frac{h}{2}(y'_{n+1} + y'_n) - \frac{h^3}{12} y'''(\xi) .$$

We recognize (5.6) as the trapezoid rule. As noted for the general case, we define

$$(5.7) \qquad y_{n+1}^{(t)} = y_n + \frac{h}{2}(f(x_{n+1}, y_{n+1}^{(p)}) + y'_n)$$

and take $y_{n+1} = y_{n+1}^{(t)}$.

As y_0 is given, we need to obtain a value only for y_1 to get started. If one wishes to avoid use of Runge-Kutta, the value of y_1 can be obtained by iterating with (5.6). The same applies if one needs to make a restart after changing the length of the step.

While the values of ξ in the error terms of (5.5) and (5.6) will scarcely ever be the same, if y''' is slowly varying then the error of $y_{n+1}^{(c)}$ will be about one fifth that of $y_{n+1}^{(p)}$ if we have iterated on (5.6) until both (5.6) and (5.3) are satisfied. In practice, the value of $y_{n+1}^{(t)}$ is close enough to this limiting value of $y_{n+1}^{(c)}$ that one can say that the error of $y_{n+1}^{(t)}$ is about one fifth that of $y_{n+1}^{(p)}$. In other words the step by step error of the new y_{n+1} is about

$$(5.8) \qquad -\frac{1}{6}(y_{n+1}^{(t)} - y_{n+1}^{(p)}) .$$

If it were not so, then one is probably using too large a value of h .

One is tempted to try for more accuracy by defining

$$y_{n+1}^{(e)} = y_{n+1}^{(t)} - \frac{1}{6} (y_{n+1}^{(t)} - y_{n+1}^{(p)}) , \tag{5.9}$$

and then taking y_{n+1} to be $y_{n+1}^{(e)}$. (We use the superscript "e" to denote "extrapolated".) This does indeed seem to give more accuracy. If one would use

$$y_{n+1}^{(c)} - \frac{1}{6} (y_{n+1}^{(c)} - y_{n+1}^{(p)}) ,$$

with the limiting value of $y_{n+1}^{(c)}$, one would have a third order method. See Ralston, 1965, p. 186. As $y_{n+1}^{(t)}$ is close to $y_{n+1}^{(c)}$, one has close to a third order method. So one should have more accuracy, but one has no way to tell how much more; there is no way to estimate the step by step error. Indeed, as we shall shortly show, numerical examples disclose a somewhat erratic behavior of $y_{n+1}^{(e)}$.

The doctrine on use of $y_{n+1}^{(e)}$ is not clear. In Ralston, 1965, on p. 186, such use is not favored. Indeed, it is there stated that it affects the stability properties, and this is also affirmed on p. 210 in Hamming, 1962. However, in Hamming, 1962, there is advocacy of more complicated schemes which make use of an equivalent of $y_{n+1}^{(e)}$. For the Adams methods, which are strongly stable, the present writer has found no evidence that use of $y_{n+1}^{(e)}$ causes stability to deteriorate, though it does indeed for some other types of predictor-corrector methods. However, the lack of any way to estimate the step by step error if one uses $y_{n+1}^{(e)}$ seems a strong point against it.

Since we have brought up the question of stability, we should note that usage of the term is not uniform. Many people, following Dahlquist, 1956, say that stability requires that all roots of a certain difference equation should be less than unity in absolute value. In Hamming, 1962, on p. 191, it is pointed out that according to that definition no method for solving $y' = ky$ could be stable for $k > 0$. So Hamming introduces (on p. 191) the notion of relative stability, in which roots can be larger than unity in absolute value provided they are less in absolute value than a certain selected root. In Definition 5.2 on p. 176 of Ralston, 1965, we find Ralston taking this as the definition of stability; what Hamming calls "relative stability" is called "stability" by Ralston. We concur with Ralston in our use of the word "stable." If a method is stable in this sense, one can safely carry out a numerical integration of indefinite extent by means of it.

It turns out that to achieve stability for predictor-corrector methods, one must maintain certain bounds for

$$h \frac{\partial f(x,y)}{\partial y} . \tag{5.10}$$

The same is claimed to be true for some Runge-Kutta methods. See Shampine and Watts, 1977, p. 270. Writers of texts on numerical analysis have been very lax about calling this to the attention of their readers, and most people probably harbor the illusion that Runge-Kutta methods are stable under all circumstances. Fortunately, the Runge-Kutta's of orders two, three, and four given in this paper happen to be relatively stable (in the sense of Hamming) in all circumstances, and hence give no trouble about stability. Far and away most people who use a Runge-Kutta use oneof these three, which is probably why almost no cases of instability have been reported when using Runge-Kutta.

A discussion of why one needs to set bounds for (5.10) to avoid instability is given in Hamming, 1962, on pp. 189-190 and in Ralston, 1965, on pp. 169-178.

Certainly, when one is using predictor-corrector methods, one should make routine estimates of $\partial f(x,y)/\partial y$ to check whether (5.10) is remaining in bounds. A numerical example of what can happen if one fails to do this will be given in Section 6. If one has two values y and $y + \varepsilon$, and ε is fairly small, then we get a good estimate by

$$\frac{\partial f(x,y)}{\partial y} \cong \frac{f(x, y+\varepsilon) - f(x, y)}{\varepsilon} . \tag{5.11}$$

For the calculations proposed, we can take $x = x_{n+1}$, $y + \varepsilon = y_{n+1}^{(t)}$, and $y = y_{n+1}^{(p)}$. Thus we can get our estimate by (5.11) without having to perform any additional function evaluations, since $f(x_{n+1}, y_{n+1}^{(t)})$ must be calculated to provide the value of y'_{n+1} required in the predictor for the next step.

The reason that $y_{n+1}^{(t)}$ is close to the $y_{n+1}^{(c)}$ that one would get by iterating the corrector, (5.6), is because $y_{n+1}^{(p)}$ is already fairly close to that $y_{n+1}^{(c)}$; the use of $y_{n+1}^{(p)}$ in (5.7) makes $y_{n+1}^{(t)}$ even closer to $y_{n+1}^{(c)}$ than $y_{n+1}^{(p)}$. If we could find a still closer approximation than $y_{n+1}^{(p)}$ to use in (5.7), we could do still better. A closer approximation is

$$(5.12) \qquad y_{n+1}^{(p)} + (y_{n+1}^{(t)} - y_{n+1}^{(p)}),$$

since this is $y_{n+1}^{(t)}$. Obviously we cannot use this until we have calculated $y_{n+1}^{(t)}$. However, with h fairly small, $y_n^{(t)} - y_n^{(p)}$ does not vary hugely from step to step. So if we define

$$(5.13) \qquad y_{n+1}^{(a)} = y_{n+1}^{(p)} + (y_n^{(t)} - y_n^{(p)}),$$

then $y_{n+1}^{(a)}$ should be closer to $y_{n+1}^{(c)}$ than $y_{n+1}^{(p)}$. (We may consider the "a" as standing for "adjusted.") This suggestion is made at the bottom of p. 201 of Hamming, 1962, but its use is there discouraged although later a slight variation of it is strongly advocated. (We shall discuss this Hamming variation later.)

If we use $y_{n+1}^{(a)}$ as an improved predictor, we would define

$$(5.14) \qquad y_{n+1}^{(ta)} = y_n + \frac{h}{2}\,(f(x_{n+1}, y_{n+1}^{(a)}) + y'_n) ,$$

and then use $y_{n+1}^{(ta)}$ for y_{n+1} instead of $y_{n+1}^{(t)}$.

Since $y_{n+1}^{(ta)}$ is closer to $y_{n+1}^{(c)}$ than $y_{n+1}^{(p)}$ or $y_{n+1}^{(t)}$, it would be better to define $y_{n+1}^{(a)}$ by

$$(5.15) \qquad y_{n+1}^{(a)} = y_{n+1}^{(p)} + (y_n^{(ta)} - y_n^{(p)}) .$$

Actually, there is trouble getting started with (5.15). If one has only y_0 and y_1, there is no way to get $y_1^{(p)}$, and so (5.15) cannot be used to get $y_2^{(a)}$. We will discuss this later. However, once one has got started, we will define

$y_{n+1}^{(a)}$ by (5.15) rather than (5.13), and use this in (5.14).

Though $y_{n+1}^{(ta)}$ is closer to $y_{n+1}^{(c)}$ than $y_{n+1}^{(t)}$ is, it is not necessarily closer to the true value of y_{n+1}. Indeed, numerical results which will be given shortly seem to support the proposition that use of $y_{n+1}^{(ta)}$ rather than $y_{n+1}^{(t)}$ results in somewhat poorer results about half of the time. However it results in appreciably better results the other half of the time. Also, its behavior is quite a bit less erratic than for $y_{n+1}^{(t)}$. So, on the whole, it seems a good idea.

To implement this, we would have to allocate an extra memory location to store $y_n^{(ta)} - y_n^{(p)}$ for use in calculating $y_{n+1}^{(a)}$ for the next step. If one is pressed for memory space, one may have to forego use of $y_{n+1}^{(a)}$. Also, there is difficulty about calculating the first instance of $y_n^{(ta)}$. If one has y_0 given, and y_1 estimated by some means, one can follow the suggestion of Hamming, 1962, on p. 206, that one would simply set $y_2^{(a)} = y_2^{(p)}$. That is, in (5.15), we take $y_1^{(ta)} - y_1^{(p)} = 0$. What this amounts to is that we use $y_2^{(t)}$ as an approximation for y_2. It may indeed be as good an approximation for y_2 as we had earlier got for y_1. Once we have done this, we can continue on by (5.14) and (5.15). Alternatively, one can generate an approximation for y_2 by the same means that we got an approximation for y_1. Then we can get $y_3^{(a)}$ by (5.13), which is close to (5.15), after which one can use (5.15) for subsequent steps.

It is again tempting to try for more accuracy by defining

$$(5.16) \qquad y_{n+1}^{(ea)} = y_{n+1}^{(ta)} - \frac{1}{6}\,(y_{n+1}^{(ta)} - y_{n+1}^{(p)}) ,$$

and then taking y_{n+1} to be $y_{n+1}^{(ea)}$. This has the same advantages and disadvantages as using $y_{n+1}^{(e)}$, but appears to be less erratic.

If we are taking y_{n+1} to be $y^{(ta)}$, then the two function evaluations that would be available to put on the right side of (5.11) would be with

$y + \varepsilon = y_{n+1}^{(ta)}$ and $y = y_{n+1}^{(a)}$. For the second order Adams method that we are considering, this should work well. However, in Hamming, 1962, on p. 207, concern is expressed that, for a high order Adams method, $y_{n+1}^{(ta)}$ and $y_{n+1}^{(a)}$ might be so close together that ε and the numerator on the right of (5.11) could be appreciably altered by round off error, so that use of (5.11) would give a poor estimate of $\partial f(x,y)/\partial y$. One should not be working close enough to the bounds for (5.10) that a sharp estimate for $\partial f(x,y)/\partial y$ is required. However, one should investigate the round off properties of his calculator to see if there is danger of a really poor answer from (5.11). If there is, one will occasionally have to use an extra function evaluation to get an extra value $f(x_{n+1},y)$ from which to estimate $\partial f(x,y)/\partial y$ by (5.11).

To give some assessment of the merits of the preceding procedures, we have calculated the relative errors that would result at $x = 6$. This has been done for the three equations $y' = y$, $y' = -y$, and $y' = -2xy^2$, all with $y(0) = 1$, and for three cases $h = 0.1$, 0.2, and 0.3. Needless to say, for $y' = -2xy^2$, the solution is $y = (1+x^2)^{-1}$. The row labelled R-K in Table 1 gives the result of the Runge-Kutta of order two that we discussed earlier. The rows labelled "t", "e", etc. refer to the cases where $y_{n+1}^{(t)}$, $y_{n+1}^{(e)}$, etc. are used for y_{n+1}. The rows labelled "tH" and "eH" will be explained shortly.

Reference to Table 1 will verify some comments which were made before. The behavior of $y_{n+1}^{(e)}$ is distinctly erratic, the most extreme case being for $y' = y$ and $h = 0.3$, where $y_{n+1}^{(e)}$ gives a poorer result than $y_{n+1}^{(t)}$. It will be noted that results using $y_{n+1}^{(ta)}$ are poorer than $y_{n+1}^{(t)}$ for $y' = y$ and better for $y' = -y$, but on the whole less erratic than for $y_{n+1}^{(t)}$.

In both Hamming, 1962, and Ralston, 1965, there is advocacy of using $y_{n+1}^{(H)}$ rather than $y_{n+1}^{(a)}$, where we define

$$(5.17) \qquad y_{n+1}^{(H)} = y_{n+1}^{(p)} + \frac{5}{6}\,(y_n^{(tH)} - y_n^{(p)}),$$

TABLE 1

Relative error at $x = 6$ for second order Adams.

		$y' = y$	$y' = -y$	$y' = -2xy^2$
$h = 0.1$	R-K	9.24×10^{-3}	-1.08×10^{-2}	-1.52×10^{-3}
	t	-3.65×10^{-3}	6.65×10^{-3}	6.12×10^{-4}
	ta	-4.88×10^{-3}	4.83×10^{-3}	4.55×10^{-4}
	tH	-4.67×10^{-3}	5.12×10^{-3}	4.81×10^{-4}
	e	7.10×10^{-4}	8.84×10^{-4}	8.91×10^{-5}
	ea	-3.26×10^{-4}	-6.12×10^{-4}	-4.23×10^{-5}
	eH	-1.47×10^{-4}	-3.74×10^{-4}	-2.09×10^{-5}
$h = 0.2$	R-K	3.39×10^{-2}	-4.76×10^{-2}	-6.56×10^{-3}
	t	-1.02×10^{-2}	3.46×10^{-2}	3.21×10^{-3}
	ta	-1.83×10^{-2}	1.71×10^{-2}	1.89×10^{-3}
	tH	-1.69×10^{-2}	1.97×10^{-2}	2.12×10^{-3}
	e	5.09×10^{-3}	7.90×10^{-3}	7.63×10^{-4}
	ea	-1.73×10^{-3}	-6.29×10^{-3}	-3.26×10^{-4}
	eH	-5.11×10^{-4}	-4.16×10^{-3}	-1.50×10^{-4}
$h = 0.3$	R-K	6.96×10^{-2}	-1.19×10^{-1}	-1.60×10^{-2}
	t	-1.47×10^{-2}	9.89×10^{-2}	9.47×10^{-3}
	ta	-3.73×10^{-2}	2.90×10^{-2}	4.61×10^{-3}
	tH	-3.31×10^{-2}	3.88×10^{-2}	5.40×10^{-3}
	e	1.54×10^{-2}	2.98×10^{-2}	2.81×10^{-3}
	ea	-3.40×10^{-3}	-2.64×10^{-2}	-9.35×10^{-4}
	eH	4.47×10^{-5}	-1.85×10^{-2}	-3.39×10^{-4}

having defined

$$(5.18) \qquad y_{n+1}^{(tH)} = y_n + \frac{h}{2}\,(f(x_{n+1}, y_{n+1}^{(H)}) + y_n') ;$$

we take y_{n+1} to be $y_{n+1}^{(tH)}$. (The superscript "H" stands for "Hamming.") This appears at the appropriate place in Table 1. Hamming further proposes taking

$$(5.19) \qquad y_{n+1}^{(eH)} = y_{n+1}^{(tH)} - \frac{1}{6}\,(y_{n+1}^{(tH)} - y_{n+1}^{(p)})$$

and using $y_{n+1}^{(eH)}$ for y_{n+1} , though Ralston frowns on this (see p. 186 of Ralston, 1965). The results of this also are shown in Table 1.

Hamming's rationale seems to be as follows. Subtracting $y_{n+1}^{(p)}$ from both sides of (5.19) gives

$$(5.20) \qquad y_{n+1}^{(eH)} - y_{n+1}^{(p)} = \frac{5}{6}\,(y_{n+1}^{(tH)} - y_{n+1}^{(p)}) .$$

So we can write (5.17) as

$$(5.21) \qquad y_{n+1}^{(H)} = y_{n+1}^{(p)} + (y_n^{(eH)} - y_n^{(p)}) .$$

If there is very little change of $y_n^{(eH)} - y_n^{(p)}$ from n to $n+1$, then we are very nearly taking $y_{n+1}^{(H)}$ to be $y_{n+1}^{(eH)}$. As expressed on p. 206 of Hamming, 1972, we "mop up" the error of $y_{n+1}^{(p)}$. Indeed, later on the same page there is the suggestion that one might just take $y_{n+1}^{(H)}$ to be y_{n+1} , and so cut down the number of function evaluations to one per step. Presumably one still has to go through the evaluation of $y_{n+1}^{(eH)}$ at each step, to have it available for use in (5.21) at the next step, but for y_{n+1}' in the next predictor we would use

$$f(x_{n+1}, y_{n+1}^{(H)}) .$$

Actually, one cannot quite hold it down to one evaluation per step, since one will occasionally need two evaluations in a step for use in (5.11). Incidentally, Hamming does not make an analysis of the stability of this proceeding. It has the disadvantage that there is no way to estimate the step by step error.

Actually, if (5.19) is to give a third order method, then $y_{n+1}^{(tH)}$ should really be $y_{n+1}^{(c)}$. For this, we should like $y_{n+1}^{(H)}$ in (5.18) to be as close as possible to $y_{n+1}^{(c)}$. But the definition (5.17), being equivalent to (5.21), is liable to make $y_{n+1}^{(H)}$ closer to $y_{n+1}^{(eH)}$ than to $y_{n+1}^{(c)}$, and the earlier procedure of using $y_{n+1}^{(ta)}$ would seem preferable. However, according to Table 1, $y_{n+1}^{(eH)}$ seems to do a shade better than $y_{n+1}^{(ea)}$. The very erratic value at $y' = y$, $h = 0.3$ for $y_{n+1}^{(eH)}$ is disquieting.

If one just stops at $y_{n+1}^{(tH)}$, and takes this to be y_{n+1}, instead of going on to $y_{n+1}^{(eH)}$ (this is what is proposed in Ralston, 1965, on p. 189), it seems better to use the simpler $y_{n+1}^{(ta)}$. From what numerical evidence we have, $y_{n+1}^{(ta)}$ is better than $y_{n+1}^{(tH)}$ in half of the cases, and worse in half.

To get started, we calculated y_1 and y_2 from the known solutions of the equations, and then (as suggested earlier) used (5.13) to calculate $y_3^{(a)}$.

If we use these methods to solve $y' = k\,y$, we get stability in the ranges shown in Table 2. This means that for stability the bounds shown in Table 2 must be satisfied by (5.10). The fact that one has stability for all positive values is surprising. This fact is not particularly useful, since for large hk the step by step errors would be so large as to render the method of little value.

TABLE 2

Stability ranges for $y' = ky$ for Adams second order.

t	$-0.6 \leq hk$	e	$-0.8 \leq hk$
ta	$-0.7 \leq hk$	ea	$-0.7 \leq hk$
tH	$-0.7 \leq hk$	eH	$-0.7 \leq hk$

Observe that in Table 2, use of extrapolated values does not diminish the region of stability. If anything, the reverse is true.

In Section 6 we will indicate how the values in Table 2 were derived.

We now turn to third order methods. We will use the Runge-Kutta

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 4k_2 + k_3) , \tag{5.22}$$

where

$$k_1 = h\, f(x_n, y_n)$$
$$k_2 = h\, f(x_n + \frac{h}{2}, y_n + \frac{k_1}{2})$$
$$k_3 = h\, f(x_n + h, y_n - k_1 + 2k_2) ;$$

see (5.6-46) on p. 199 of Ralston, 1965, or 25.5.8 on p. 896 of Abramowitz and Stegun, 1964.

The third order Adams is given by

$$y^{(p)}_{n+1} = y_n + \frac{h}{12}(23y'_n - 16y'_{n-1} + 5y'_{n-2}) + \frac{9h^4}{24} y^{(iv)}(\xi) \tag{5.23}$$

$$y^{(c)}_{n+1} = y_n + \frac{h}{12}(5y'_{n+1} + 8y'_n - y'_{n-1}) - \frac{h^4}{24} y^{(iv)}(\xi) . \tag{5.24}$$

Analogously to the second order Adams, we set

$$y^{(t)}_{n+1} = y_n + \frac{h}{12}(5f(x_{n+1}, y^{(p)}_{n+1}) + 8y'_n - y'_{n-1}) \tag{5.25}$$

$$y^{(e)}_{n+1} = y^{(t)}_{n+1} - \frac{1}{10}(y^{(t)}_{n+1} - y^{(p)}_{n+1}) \tag{5.26}$$

$$y^{(a)}_{n+1} = y^{(p)}_{n+1} + (y^{(ta)}_n - y^{(p)}_n) \tag{5.27}$$

$$y^{(ta)}_{n+1} = y_n + \frac{h}{12}(5f(x_{n+1}, y^{(a)}_{n+1}) + 8y'_n - y'_{n-1}) \tag{5.28}$$

$$y^{(ea)}_{n+1} = y^{(ta)}_{n+1} - \frac{1}{10}(y^{(ta)}_{n+1} - y^{(p)}_{n+1}) \tag{5.29}$$

$$y^{(H)}_{n+1} = y^{(p)}_{n+1} + \frac{9}{10}(y^{(tH)}_n - y^{(p)}_n) \tag{5.30}$$

$$y^{(tH)}_{n+1} = y_n + \frac{h}{12}(5f(x_{n+1}, y^{(H)}_{n+1}) + 8y'_n - y'_{n-1}) \tag{5.31}$$

$$y^{(eH)}_{n+1} = y^{(tH)}_{n+1} - \frac{1}{10}(y^{(tH)}_{n+1} - y^{(p)}_{n+1}) . \tag{5.32}$$

TABLE 3

Relative error at $x = 6$ for third order Adams.

		$y' = y$	$y' = -y$	$y' = -2xy^2$
$h = 0.1$	R-K	2.31×10^{-4}	2.71×10^{-4}	3.31×10^{-5}
	t	-1.50×10^{-4}	-3.80×10^{-4}	-2.06×10^{-5}
	ta	-2.32×10^{-4}	-2.47×10^{-4}	-1.82×10^{-5}
	tH	-2.24×10^{-4}	-2.60×10^{-4}	-1.85×10^{-5}
	e	5.90×10^{-5}	-7.98×10^{-5}	-9.36×10^{-7}
	ea	-1.50×10^{-5}	3.82×10^{-5}	1.31×10^{-6}
	eH	-7.38×10^{-6}	2.69×10^{-5}	1.04×10^{-6}
$h = 0.2$	R-K	1.70×10^{-3}	2.35×10^{-3}	3.03×10^{-4}
	t	-5.96×10^{-4}	-4.38×10^{-3}	-1.27×10^{-4}
	ta	-1.62×10^{-3}	-1.71×10^{-3}	-1.76×10^{-4}
	tH	-1.51×10^{-3}	-1.95×10^{-3}	-1.76×10^{-4}
	e	8.14×10^{-4}	-1.49×10^{-3}	9.04×10^{-6}
	ea	-1.17×10^{-4}	8.73×10^{-4}	-2.77×10^{-5}
	eH	-1.85×10^{-5}	6.58×10^{-4}	-2.78×10^{-5}
$h = 0.3$	R-K	5.30×10^{-3}	8.56×10^{-3}	1.19×10^{-3}
	t	-4.58×10^{-4}	-2.06×10^{-2}	-3.48×10^{-5}
	ta	-4.49×10^{-3}	-3.70×10^{-3}	-8.99×10^{-4}
	tH	-4.06×10^{-3}	-5.15×10^{-3}	-4.05×10^{-4}
	e	3.56×10^{-3}	-8.84×10^{-3}	2.52×10^{-4}
	ea	-1.33×10^{-4}	6.04×10^{-3}	-4.05×10^{-4}
	eH	2.67×10^{-4}	4.77×10^{-3}	-3.97×10^{-4}

To get started, we can estimate y_1 and y_2 by some means and then take $y_3^{(a)} = y_n^{(p)}$; this amounts to approximating y_3 by $y_3^{(t)}$, and is what was done to get the values in Table 3. Alternatively one can estimate y_1, y_2, and y_3 by some means, and then take either

$$y_4^{(a)} = y_4^{(p)} + (y_3^{(t)} - y_3^{(p)})$$

or

$$y_4^{(H)} = y_4^{(p)} + \frac{9}{10}(y_3^{(t)} - y_3^{(p)}) ;$$

subsequently one uses either (5.27) or (5.30).

Some numerical results are given in Table 3. The column headed $y' = -2xy^2$ is quite erratic. This bears out the remark at the top of p. 210 in Hamming, 1962, that the corresponding solution is often troublesome to approximate by polynomials.

For the equation $y' = ky$, we get stability in the ranges shown in Table 4. These bounds should be satisfied by (5.10). For the case $h = 0.3$ for $y' = -2xy^2$, the bounds are not satisfied by (5.10) for a region near $x = 1$.

TABLE 4

Stability ranges for $y' = ky$ for Adams third order.

t	$-0.8 \leq hk$	e	$-0.9 \leq hk$
ta	$-0.5 \leq hk$	ea	$-0.5 \leq hk$
tH	$-0.5 \leq hk$	eH	$-0.5 \leq hk$

However, we got through the region of instability in two or three steps, which were not enough for the instability to build up appreciably. For most of the range of integration, (5.10) was well within the bounds of stability.

As with the second order Adams method, use of extrapolated values does not diminish the region of stability.

To get some feeling how erratic some of the values are in Table 3, note that R-K, t, ta, and tH are supposed to be third order methods. Thus the error for $h = 0.2$ should be 8 times that for $h = 0.1$, and the error for $h = 0.3$ should be 27 times that for $h = 0.1$. This works out reasonably well except for t .

As pointed out on p. 186 of Ralston, 1965, use of

$$y_{n+1} = y^{(c)}_{n+1} - \frac{1}{10}\,(y^{(c)}_{n+1} - y^{(p)}_{n+1}) \tag{5.33}$$

should give a fourth order method. As $y^{(t)}_{n+1}$, $y^{(ta)}_{n+1}$, and $y^{(tH)}_{n+1}$ are close to $y^{(c)}_{n+1}$, use of (5.26), (5.29), and (5.32) means that e, ea, and eH should be close to fourth order methods. So we look for the error for $h = 0.2$ to be 16 times that for $h = 0.1$, and the error for $h = 0.3$ to be 81 times that for $h = 0.1$. None of e, ea, or eH comes very close to such behavior. Nor can one count on a striking increase in accuracy from using extrapolated values. In two cases eH is only barely better than tH, and in one case ea is poorer than ta . Considering that there is no way to estimate step by step error when one is using extrapolated methods, they should probably not be used.

For the fourth order Runge-Kutta, consult (4.3) and (4.4), which are generalized versions. For the fourth order Adams method (see Conte and de Boor, 1972, p. 342 and p. 351), we set

$$y^{(p)}_{n+1} = y_n + \frac{h}{24}\,(55y'_n - 59y'_{n-1} + 37y'_{n-2} - 9y'_{n-3}) + \frac{251h^5}{720}\,y^{(v)}(\xi) \tag{5.34}$$

$$y^{(c)}_{n+1} = y_n + \frac{h}{24}\,(9y'_{n+1} + 19y'_n - 5y'_{n-1} + y'_{n-2}) - \frac{19h^5}{720}\,y^{(v)}(\xi). \tag{5.35}$$

These give

$$y^{(t)}_{n+1} = y_n + \frac{h}{24}\,(9f(x_{n+1}, y^{(p)}_{n+1}) + 19y'_n - 5y'_{n-1} + y'_{n-2}) \tag{5.36}$$

$$y^{(e)}_{n+1} = y^{(t)}_{n+1} - \frac{19}{270}\,(y^{(t)}_{n+1} - y^{(p)}_{n+1}) \tag{5.37}$$

$$y^{(a)}_{n+1} = y^{(p)}_{n+1} + (y_n^{(ta)} - y_n^{(p)}) \tag{5.38}$$

TABLE 5

Relative error at $x = 6$ for fourth order Adams .

		$y' = y$	$y' = -y$	$y' = -2xy^2$
$h = 0.1$	R-K	4.62×10^{-6}	-5.44×10^{-6}	-8.10×10^{-7}
	t	-7.65×10^{-6}	2.73×10^{-5}	9.87×10^{-7}
	ta	-1.39×10^{-5}	1.61×10^{-5}	1.02×10^{-6}
	tH	-1.35×10^{-5}	1.69×10^{-5}	1.02×10^{-6}
	e	5.11×10^{-6}	7.50×10^{-6}	-5.63×10^{-7}
	ea	-8.10×10^{-7}	-2.88×10^{-6}	-5.41×10^{-7}
	eH	-4.32×10^{-7}	-2.17×10^{-6}	-5.40×10^{-7}
$h = 0.2$	R-K	6.77×10^{-5}	-9.46×10^{-5}	-1.37×10^{-5}
	t	-2.98×10^{-5}	6.93×10^{-4}	-5.46×10^{-5}
	ta	-1.81×10^{-4}	2.18×10^{-4}	5.65×10^{-6}
	tH	-1.70×10^{-4}	2.48×10^{-4}	2.11×10^{-6}
	e	1.34×10^{-4}	2.94×10^{-4}	-4.97×10^{-5}
	ea	-7.66×10^{-6}	-1.44×10^{-4}	4.80×10^{-6}
	eH	2.98×10^{-6}	-1.15×10^{-4}	1.60×10^{-6}
$h = 0.3$	R-K	3.16×10^{-4}	-5.21×10^{-4}	-7.19×10^{-5}
	t	1.71×10^{-4}	5.30×10^{-3}	-9.46×10^{-4}
	ta	-6.85×10^{-4}	5.39×10^{-4}	5.59×10^{-4}
	tH	-6.20×10^{-4}	8.28×10^{-4}	4.55×10^{-4}
	e	8.43×10^{-4}	2.74×10^{-3}	-6.92×10^{-4}
	ea	3.70×10^{-5}	-1.61×10^{-3}	6.37×10^{-4}
	eH	9.81×10^{-5}	-1.35×10^{-3}	5.48×10^{-4}

(5.39) $$y_{n+1}^{(ta)} = y_n + \frac{h}{24}(9f(x_{n+1}, y_{n+1}^{(a)}) + 19y_n' - 5y_{n-1}' + y_{n-2}')$$

(5.40) $$y_{n+1}^{(ea)} = y_{n+1}^{(ta)} - \frac{19}{270}(y_{n+1}^{(ta)} - y_{n+1}^{(p)})$$

(5.41) $$y_{n+1}^{(H)} = y_{n+1}^{(p)} + \frac{251}{270}(y_n^{(tH)} - y_n^{(p)})$$

(5.42) $$y_{n+1}^{(tH)} = y_n + \frac{h}{24}(9f(x_{n+1}, y_{n+1}^{(H)}) + 19y_n' - 5y_{n-1}' + y_{n-2}')$$

(5.43) $$y_{n+1}^{(eH)} = y_{n+1}^{(tH)} - \frac{19}{270}(y_{n+1}^{(tH)} - y_{n+1}^{(p)}) .$$

The matter of getting started can be handled as in the second and third order Adams methods.

Some numerical results are given in Table 5. The extrapolated values are distinctly erratic, as is also $y_{n+1}^{(t)}$. However, $y_{n+1}^{(ta)}$ behaves fairly well. The excellent results from the Runge-Kutta are worth remarking.

For the equation $y' = ky$, we get stability in the ranges shown in Table 6. These bounds should be satisfied by (5.10). For $h = 0.3$ for $y' = -2xy^2$, this was not the case for a short interval, not long enough to cause any trouble.

TABLE 6

Stability ranges for $y' = ky$ for Adams fourth order.

t	$-0.6 \leq hk$	e	$-0.6 \leq hk$
ta	$-0.4 \leq hk$	ea	$-0.4 \leq hk$
tH	$-0.4 \leq hk$	eH	$-0.4 \leq hk$

Use of extrapolated values does not diminish the region of stability.

6. Milne type predictor-correctors. In Southard and Yowell, 1952, is given

$$(6.1) \qquad y_{n+1}^{(p)} = -4y_n + 5y_{n-1} + h(4y'_n + 2y'_{n-1}) + \frac{h^4}{6} y^{(iv)}(\xi)$$

$$(6.2) \qquad y_{n+1}^{(c)} = y_n + \frac{h}{12}(5y'_{n+1} + 8y'_n - y'_{n-1}) - \frac{h^4}{24} y^{(iv)}(\xi) .$$

One will recognize that the corrector is the same as for the third order Adams, to wit (5.24). So the formula for $y_{n+1}^{(t)}$ would be the same as (5.25). We would have

$$(6.3) \qquad y_{n+1}^{(e)} = y_{n+1}^{(t)} - \frac{1}{5}(y_{n+1}^{(t)} - y_{n+1}^{(p)}) ,$$

and the other formulas analogously.

This predictor-corrector has some good features. It is third order, but has modest memory requirements, and is easy to get started, or get restarted after changing the length of the step. However, it behaves poorly as to stability, as can be seen from Table 7 and Table 8. The fact that all positive values of hk are excluded for ea and eH is startling.

Because the range of stability for $y_{n+1}^{(t)}$ is so limited, one finds (3.10) outside that range in the case $h = 0.3$ and $y' = -2xy^2$ for an extended period, and the solution really blows up. Already at $x = 3$, one gets a negative value for y, after which the errors compound catastrophically. Before reaching $x = 6$, the calculator stops on an overflow, because the numbers are too large for its capacity.

If one had been monitoring (5.10), this instability could have been avoided by using a smaller h through the critical region.

The analysis of the stability for $y_{n+1}^{(t)}$ and $y_{n+1}^{(e)}$ would be worthy of special attention by anyone interested in stability. Neither the usual Dahlquist criterion of stability nor the Hamming-Ralston criterion for relative stability is applicable, and a special definition had to be contrived. Never mind the details. The reader can trust that if (5.10) stays too long outside the bounds

TABLE 7

Relative error at $x = 6$ for Southard and Yowell.

		$y' = y$	$y' = -y$	$y' = -2xy^2$
$h = 0.1$	R-K	2.31×10^{-4}	2.71×10^{-4}	3.31×10^{-5}
	t	-1.64×10^{-4}	-4.18×10^{-4}	-1.50×10^{-5}
	ta	-2.32×10^{-4}	-2.48×10^{-4}	-2.01×10^{-5}
	tH	-2.17×10^{-4}	-2.73×10^{-4}	-2.09×10^{-5}
	e	1.18×10^{-5}	-1.52×10^{-5}	-1.82×10^{-7}
	ea	unstable	9.43×10^{-6}	5.77×10^{-7}
	eH	unstable	5.05×10^{-6}	4.16×10^{-7}
$h = 0.2$	R-K	1.70×10^{-3}	2.35×10^{-3}	3.03×10^{-4}
	t	-8.99×10^{-4}	-7.16×10^{-3}	4.54×10^{-3}
	ta	-1.63×10^{-3}	-1.75×10^{-3}	7.17×10^{-5}
	tH	-1.45×10^{-3}	-2.20×10^{-3}	-1.34×10^{-4}
	e	1.70×10^{-4}	-2.84×10^{-4}	4.00×10^{-6}
	ea	unstable	1.79×10^{-4}	4.03×10^{-5}
	eH	unstable	1.09×10^{-4}	1.05×10^{-5}
$h = 0.3$	R-K	5.30×10^{-3}	8.56×10^{-3}	1.19×10^{-3}
	t	-2.09×10^{-3}	-2.79×10^{-1}	unstable
	ta	-4.66×10^{-3}	unstable	1.30×10^{-1}
	tH	-4.00×10^{-3}	unstable	3.45×10^{-2}
	e	7.79×10^{-4}	-1.72×10^{-3}	1.59×10^{-4}
	ea	unstable	unstable	unstable
	eH	unstable	unstable	1.04×10^{-1}

TABLE 8

Stability ranges for $y' = ky$ for Southard and Yowell.

t	$-0.31 \leq hk$	e	$-0.7 \leq hk$
ta	$-0.25 \leq hk$	ea	$-0.25 \leq hk < 0$
tH	$-0.28 \leq hk$	eH	$-0.28 \leq hk < 0$

given in Table 8, there will be trouble; see the case $h = 0.3$ for $y' = -2xy^2$ for example.

In Hamming, 1959, on p. 47, concern is expressed about the stability of Southard and Yowell, and it is proposed to remedy the situation by using a different predictor

$$(6.4) \qquad y_{n+1}^{(p)} = y_n + y_{n-1} - y_{n-2} + 2h(y'_n - y'_{n-1}) + \frac{h^4}{3} y^{(iv)}(\xi) .$$

This appreciably increases the number of memory locations needed and requires more starting values to get started or restarted, thus cancelling out the two good features of Southard and Yowell. It does appear to improve the stability; for $y_{n+1}^{(t)}$ it is increased to $-0.5 \leq hk$. However, this is considerably less than for the Adams method of order three. If we compare the Adams of order three with what results from (6.4) we find that both are of order three, both need two extra starting values to get started or restarted, but the Adams is considerably more stable and requires fewer memory locations.

In Hamming, 1959, it was proposed to render the classical Milne predictor-corrector stable by using a different corrector, namely

$$(6.5) \qquad y_{n+1}^{(c)} = \frac{1}{8} (9y_n - y_{n-2} + 3h (y'_{n+1} + 2y'_n - y'_{n-1})) - \frac{h^5}{40} y^{(v)}(\xi) ;$$

the old predictor, (4.5), was retained. The results of this are shown in Table 9 and Table 10. In Table 9 there is no column $y' = -2xy^2$ because the memory requirements exceeded the capacity of the HP-65 with which the calculations of this paper were performed. Although there is a region of stability, it is less than for the Adams method of comparable order, namely order four. All in all, the Hamming method is not a contender for use with hand held calculators.

From Ralston, 1965, one would get the impression that Hamming's method is very superior. It is $y_{n+1}^{(tH)}$ that gets the stamp of approval on p. 189 of Ralston, 1965. Actually, $y_{n+1}^{(ta)}$ would be simpler, and gives about the same results. We do not believe that $y_{n+1}^{(tH)}$ is enough superior to $y_{n+1}^{(ta)}$ to be

TABLE 9

Relative error at $x = 6$ for Hamming.

		$y' = y$	$y' = -y$
$h = 0.1$	R-K	4.62×10^{-6}	-5.44×10^{-6}
	t	-8.63×10^{-6}	4.55×10^{-5}
	ta	-1.68×10^{-5}	2.10×10^{-5}
	tH	-1.61×10^{-5}	2.24×10^{-5}
	e	4.24×10^{-6}	7.52×10^{-6}
	ea	-8.10×10^{-7}	-2.55×10^{-6}
	eH	-4.32×10^{-7}	-1.92×10^{-6}
$h = 0.2$	R-K	6.77×10^{-5}	-9.46×10^{-5}
	t	-3.87×10^{-5}	2.37×10^{-3}
	ta	-2.07×10^{-4}	2.82×10^{-4}
	tH	-1.92×10^{-4}	3.35×10^{-4}
	e	1.07×10^{-4}	3.73×10^{-4}
	ea	-8.49×10^{-6}	-1.15×10^{-4}
	eH	1.60×10^{-6}	-9.44×10^{-5}
$h = 0.3$	R-K	3.16×10^{-4}	-5.21×10^{-4}
	t	1.05×10^{-4}	unstable
	ta	-7.53×10^{-4}	6.78×10^{-4}
	tH	-6.75×10^{-4}	1.08×10^{-3}
	e	6.63×10^{-4}	5.90×10^{-3}
	ea	2.32×10^{-5}	-1.10×10^{-3}
	eH	8.02×10^{-5}	-9.56×10^{-4}

TABLE 10

Stability ranges for $y' = ky$ for Hamming.

t	$-0.26 \le hk$	e	$-0.38 \le hk$
ta	$-0.37 \le hk$	ea	$-0.38 \le hk$
tH	$-0.38 \le hk$	eH	$-0.39 \le hk$

worth the extra trouble of programming and the longer running time that it entails.

Despite Ralston's endorsement of Hamming, Enright and Hull, 1976, give it a very low rating on pp. 954-955. Interestingly enough, though Hamming invented the method, and devotes a lot of space discussing it in Hamming, 1962, he finally (in pp. 206-210) seems to favor something related to an Adams method.

On p. 46 of Hamming, 1959, the stability for $y_{n+1}^{(eH)}$ is claimed to be good down to about -0.65. As this disagrees with the value given in Table 10, we will justify the value in Table 10.

For Hamming's method, we have

$$(6.6) \qquad y_{n+1}^{(H)} = y_{n+1}^{(p)} + \frac{112}{121}(y_n^{(tH)} - y_n^{(p)})$$

$$(6.7) \qquad y_{n+1}^{(eH)} = y_{n+1}^{(tH)} - \frac{9}{121}(y_{n+1}^{(tH)} - y_{n+1}^{p)}) .$$

Subtracting $y_{n+1}^{(p)}$ from both sides of (6.7) gives

$$(6.8) \qquad y_{n+1}^{(eH)} - y_{n+1}^{(p)} = \frac{112}{121}(y_{n+1}^{(tH)} - y_{n+1}^{(p)}) .$$

So, by (6.6)

$$(6.9) \qquad y_{n+2}^{(H)} = y_{n+2}^{(p)} + (y_{n+1}^{(eH)} - y_{n+1}^{(p)}) .$$

As the entries that are accepted are those with a superscript (eH), we shall simplify the notation by omitting this. Then by (4.5) and (6.9), we see that for the equation $y' = ky$, we have

$$(6.10) \qquad y_{n+2}^{(H)} = \{y_{n-2} + \frac{4hk}{3}(2y_{n+1} - y_n + 2y_{n-1})\}$$
$$+ (y_{n+1} - \{y_{n-3} + \frac{4hk}{3}(2y_n - y_{n-1} + 2y_{n-2})\}).$$

This simplifies to

$$(6.11) \qquad y_{n+2}^{(H)} = y_{n+1} + y_{n-2} - y_{n-3} + \frac{4hk}{3}(2y_{n+1} - 3y_n + 3y_{n-1} - 2y_{n-2}).$$

We have, by (6.5),

$$y_{n+2}^{(tH)} = \frac{1}{8}\,(9y_{n+1} - y_{n-1} + 3hk(y_{n+2}^{(H)} + 2y_{n+1} - y_n)). \tag{6.12}$$

So

$$\begin{aligned} y_{n+2}^{(tH)} = \frac{1}{8}\,(9y_{n+1} - y_{n-1} &+ hk(9y_{n+1} - 3y_n + 3y_{n-2} - 3y_{n-3}) \\ &+ (hk)^2(8y_{n+1} - 12y_n + 12y_{n-1} - 8y_{n-2})). \end{aligned} \tag{6.13}$$

By (6.7)

$$y_{n+2} = \frac{112}{121}\,y_{n+2}^{(tH)} + \frac{9}{121}\,y_{n+2}^{(p)}\ . \tag{6.14}$$

By (4.5) and (6.13), this gives

$$\begin{aligned} y_{n+2} = \frac{1}{121}\,(126y_{n+1} - 14y_{n-1} &+ 9y_{n-2} \\ &+ hk\,(150y_{n+1} - 54y_n + 24y_{n-1} + 42y_{n-2} - 42y_{n-3}) \\ &+ (hk)^2(112y_{n+1} - 168y_n + 168y_{n-1} - 112y_{n-2})). \end{aligned} \tag{6.15}$$

The solution of this difference equation is

$$y_n = \sum_{i=1}^{5} C_i\,\rho_i^n\ , \tag{6.16}$$

where the ρ_i are the roots of the equation

$$\begin{aligned} 121\rho^5 - (126 + 150hk + 112(hk)^2)\rho^4 &+ (54hk + 168(hk)^2)\rho^3 \\ &+ (14 - 24hk - 168(hk)^2)\rho^2 - (9 + 42hk - 112(hk)^2)\rho + 42hk = 0\ . \end{aligned} \tag{6.17}$$

If we take $hk = -0.4$, we get

$$121\rho^5 - 83.92\rho^4 + 5.28\rho^3 - 3.28\rho^2 + 25.72\rho - 16.8 = 0\ . \tag{6.18}$$

The polynomial has the factors

$$\rho - 0.67053\ \ 10701 \tag{6.19}$$

$$\rho^2 + 0.92764\ 94248\rho + 0.45383\ \ 23357 \tag{6.20}$$

$$\rho^2 - 0.95067\ \ 20739\rho + 0.45625\ \ 70288. \tag{6.21}$$

The zeros of (6.20) have absolute value

0.67367 07918

and the zeros of (6.21) have absolute value

$$0.67546\ 80072\ .$$

For large n, the powers of these quantities will predominate, and the values of y_n will jump about very erratically.

For $hk = -0.39$, the factor corresponding to (6.19) will produce the dominant ρ, and (6.16) will approximate e^{hkn}, as it should.

Amongst the disadvantages of the Hamming method for hand held calculators is its excessive requirement for memory locations. To improve this, one could change the predictor to

$$(6.22) \qquad y_{n+1}^{(p)} = -9y_n + 9y_{n-1} + y_{n-2} + h(6y_n' + 6y_{n-1}') + \frac{h^5}{10} y^{(v)}(\xi)\ .$$

When used with the Hamming corrector, (6.5), this still gives a fourth order method. However, it requires two fewer memory locations. Also, it requires fewer starting values to get a start or restart, being in this respect also superior to the Adams method of order four.

This predictor is one of a set which, in a footnote on p. 171 of Hamming, 1962, is dismissed as not being worth consideration. The combination of (6.22) and (6.5) does turn out to have very poor stability characteristics. For $y_{n+1}^{(t)}$ one must have $-0.13 \le hk$, and for $y_{n+1}^{(ta)}$ one must have $-0.19 \le hk \le 0.28$. At this point, we became disheartened, and did not check out the other cases.

7. Conclusions. Despite heroic efforts to remedy various faults of the Milne type predictor-correctors, they are still decidedly inferior to the Adams type. One would prefer if the Adams type results were less erratic than they are, but if the user is diligent to keep (5.10) within bounds, it appears that one can use the Adams methods safely, and without running into excessive calculation times. An occasional "look ahead" to plan the progress of the calculation is advisable.

Runge-Kutta methods have many advantages and, if it happens that $f(x,y)$ can be evaluated quickly, they should be given serious consideration. Although the three Runge-Kutta's given in this paper are relatively stable under all circumstances, so that one can use them without any concern about stability, one must still monitor the size of h carefully, since if one allows the step by step errors to become excessive the final results can be almost without meaning.

REFERENCES

M. Abramowitz and I. A. Stegun, "Handbook of mathematical functions," Applied Mathematics Series 55, Natl. Bureau of Standards, 1964.

R. Bulirsch and R. Stoer, "Numerical treatment of ordinary differential equations by extrapolation methods," Numer. Math., vol. 8 (1966), pp. 1-13.

S. D. Conte and Carl de Boor, "Elementary numerical analysis," second edition, McGraw-Hill Book Co., 1972.

G. Dahlquist, "Convergence and stability in the numerical integration of ordinary differential equations," Math. Scand., vol. 4 (1956), pp. 33-53.

W. H. Enright and T. E. Hull, "Test results on initial value methods for non-stiff ordinary differential equations," SIAM Jour. Num. Anal., vol. 13 (1976), pp. 944-961.

E. Fehlberg, "Classical fifth-, sixth-, seventh-, and eighth-order Runge-Kutta formulas with stepsize control," NASA Tech. Report No. 287, 1968, Huntsville.

E. Fehlberg, "Klassische Runge-Kutta-Formeln fünfter und siebenter Ordnung mit Schrittweiten-Kontrolle," Computing, vol. 4 (1969), pp. 93-106.

E. Fehlberg, "Klassische Runge-Kutta-Formeln vierter und niedriger Ordnung mit Schrittweiten-Kontrolle und ihre Anwendung auf Wärmeleitungsprobleme," Computing, vol. 6 (1970), pp. 61-71.

W. B. Gragg, "On extrapolation algorithms for ordinary initial value problems," SIAM Jour. Num. Anal., vol. 2 (1965), pp. 384-403.

R. W. Hamming, "Stable predictor-corrector methods for ordinary differential equations," Jour. of ACM, vol. 6 (1959), pp. 37-47.

R. W. Hamming, "Numerical methods for scientists and engineers," McGraw-Hill Book Co., 1962.

Peter Henrici, "Discrete variable methods in ordinary differential equations," John Wiley and Sons, 1962.

Peter Henrici, "Computational analysis with the HP-25 pocket calculator," John Wiley and Sons, 1977.

T. E. Hull, W. H. Enright, B. M. Fellen, and A. E. Sedgwick, "Comparing numerical methods for ordinary differential equations," SIAM Jour. Num. Anal., vol. 9 (1972), pp. 603-637.

W. E. Milne, "Numerical solution of differential equations," John Wiley and Sons, 1953.

Anthony Ralston, "A first course in numerical analysis," McGraw-Hill Book Co., 1965.

J. Barkley Rosser, "A Runge-Kutta for all seasons," SIAM Review, vol. 9 (1967), pp. 417-452.

L. F. Shampine and H. A. Watts, "The art of writing a Runge-Kutta Code, Part I," in "Mathematical Software III," ed. by John R. Rice, Academic Press, 1977.

T. H. Southard and E. C. Yowell, "An alternative 'predictor-corrector' process," Math. Tables and Other Aids to Comp., vol. 6 (1952), pp. 253-4.

JBR/db

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
1848 6

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER
9

4. TITLE (and Subtitle)
SOLVING DIFFERENTIAL EQUATIONS ON A HAND HELD PROGRAMMABLE CALCULATOR

5. TYPE OF REPORT & PERIOD COVERED
Summary Report, 2 no specific reporting period

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
10 J. Barkley Rosser

8. CONTRACT OR GRANT NUMBER(s)
15 DAAG29-75-C-0024

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Mathematics Research Center, University of Wisconsin
610 Walnut Street
Madison, Wisconsin 53706

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
7 - Numerical Analysis

11. CONTROLLING OFFICE NAME AND ADDRESS
U. S. Army Research Office
P.O. Box 12211
Research Triangle Park, North Carolina 27709

12. REPORT DATE
11 April 1978

13. NUMBER OF PAGES
40

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
12 45p.

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited.
14 MRC-TSR-1848

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Runge-Kutta
predictor-corrector
numerical stability

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Most scientists who occasionally have to solve numerically a differential equation now own a hand held programmable calculator which will very often be adequate. Since hand held calcluators are slow, there is particular need to keep the number of function evaluations to a minimum. At first thought, this would seem to rule out use of Runge-Kutta methods, but recent developments ~~such as those by Fehlberg (mostly unknown except to specialists)~~, may make them competitive after all. In the area of predictor-corrector methods, some variations make excessive use of memory locations for a hand held calculator. An analysis

→ next page

221200

LB

of such matters is made in order to advise as to good procedures to follow, including alerting the solver to methods that are seldom taught in numerical analysis courses (where the emphasis is on the use of large fast computers).